Electronic Health Record Data Model Optimized for Knowledge Discovery

Shaker H. El-Sappagh[1], Samir El-Masri[2], A. M. Riad[3], Mohammed Elmogy[4]

[1]Collage of Science, King Saud University, Saudi Arabia

[2]School of Computer and Information Sciences, King Saud University, Saudi Arabia

[1, 3,4]Faculty of Computers & Information, Mansoura University, Egypt

Abstract

Database is a core component the Electronic Health Record (EHR) system, and creating a data model for that database is challenging due to the EHR system's special nature. Because of complexity, spatial, sparseness, interrelation, temporal, heterogeneity, and fast evolution of EHR data, modeling its database is complex process. This paper tried to build dynamic, complete and stable data model for EHR database. There are a little work and standards in this aspect because of its difficulty. We will use object relational modeling approach and entity attribute value with classes and relationships to build the model. This design facilitates and enhances the operations of data mining and decision support which is integrating component of an EHR system.

Keywords: *Electronic Health Record, database, health informatics, data mining, database design.*

1. INTRODUCTION

The Electronic Health Record (HER) is a longitudinal electronic record of patient health information produced by encounters in one or more care settings. Included in this information are patient demographics, progress notes, problems, medications, vital signs, past medical history, immunizations, laboratory data and radiology reports. The EHR is an integration of patient information systems [1]. The EHR also includes the network that links these systems, databases, interfaces, physician order entry, electronic communication systems, and the clinical workstations (ISO TC 215 [35], ISO/TR 20514:2005 [36]). EHR systems are developed to improve quality of care [5], reduce organizational expense, produce a data stream for electronic billing, and provide electronic support for secondary users (payers, policymakers and researchers) [2].

To support the previous functions, HER system contains a mix of highly structured data. These data include medical, security and privacy, legal and financial data. The most critical component is the EHR database. Having a highly structured, dynamic, complete and consistent database is important to achieve EHR objectives. This database has different architectures if EHR is centralized, distributed or hybrid. Healthcare systems moved from isolated systems in hospitals towards solutions which include multiple healthcare professionals and institutions, interoperability and information sharing. It becomes crucial to allow all relevant patients clinical data to be available at anytime, anywhere and to improve the quality and delivery of healthcare. The interoperability can be achieved using two solutions:

1- Building EHR structure and content in each hospital by unified technology and terminology standards. Although this solution facilitate the interoperability, it is difficult to apply because many existing systems use different standards and technologies.

2- Building EHR structure and content locally in each hospital using any technology and standards and use the interoperability standards for EHRs which are offered by the world leading standard bodies as: European Committee for Standardization (CEN) [39], International Organization for Standardization (ISO) [37], Health Level Seven (HL7) [14] and Digital Imaging and Communications in Medicine (DICOM) [38]. In this paper, we will use the second solution and build a data model for the EHR in local hospital and leave the connectivity and interoperability burden to the available standards [15, 16, 17]. The complexity and the rapid evolution and expansion of the clinical domain make development and maintenance of clinical databases difficult. Whenever new data types are introduced or existing types are modified in a conventional relational database system, the physical design of the database must be changed accordingly. In this paper, we try to build a temporal, dynamic, and generic clinical data model which solve the above challenges. The model uses object-relational model and use Entity-Attribute-Value with Classes and Relationship (EAV/CR) [8]. Moreover, it is compatible with ASTM E 1384 [3, 6] and CEN/TC251 preENV 12381 [4] standards. The paper is organized as follows: Section 2 discusses related works. Problem statement is discusses in section 3. Section 4 is the proposed data model. The paper is concluded in section 5.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

2. RELATED WORK

In this section, we go through some background about essential aspects of health informatics related to EHR and database design.

2.1 Data Models

There are many design methodologies for database as range from hierarchical model to object/relational model and object-oriented model. Each modeling approach has its advantages and disadvantages. Object model is suitable for the design of complex databases as EHR but its Database Management System (DBMS) is not popular and its query languages are complex. Relational model [7] is the most applicable model. However the sparseness, complex data types, and speed evolution of clinical data make the conventional design approach costly. Extending the relational model by allowing the storage of objects and solving the sparseness and evolution of data can permit the development of robust and generic model. As will be shown in section 4.1, this approach is EAV/CR [9, 10].

2.2 EHR Database Architectures

There are two architectures of EHR, basic and universal. The *basic architecture* is built inside one organization (i.e hospital) and connected to all hospital information systems as shown in figure 1 [11].

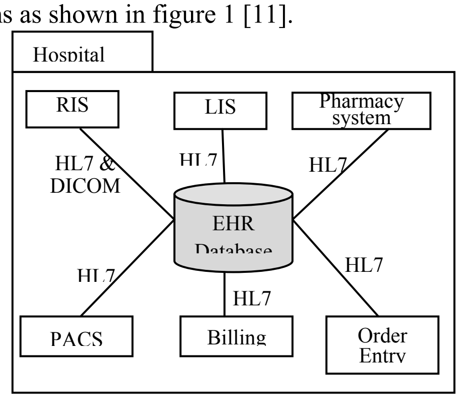

Fig. 1: Basic EHR architecture

In figure 1, the EHR database is centralized and collect data from all operating healthcare systems in the hospital as radiology information system (RIS) and others. *The universal EHR architecture has two categories.* In the first one, it may be centralized system as shown in figure 2. The centralized EHR database will be fed from different healthcare systems such as hospitals. The feed will be messaging based on standard as HL7 V3.0 RIM (Reference Information Model) and a pre-agreed data set [37]. In the second category, it can be a distributed system as shown in figure 3. In this architecture each healthcare organization has its own EHR with its own data model and with its own terminology standards.

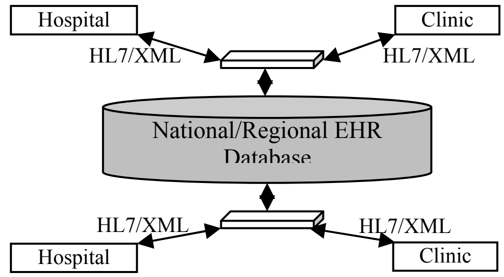

Fig. 2: Centralized universal EHR architecture

These are autonomous and heterogeneous systems developed under different objectives, and consequently with diverse data models, platforms, standards and semantics. In the best conditions, they could follow a standard, but different ones could be selected for each system. These conflicts can be classified into three levels:

(1) Semantic: the information models or database schemas are different.

(2) Functional: the interfaces to manage data are different.

(3) Instance: the information about the same real entity could differ from one system to another. The interoperability standards as HL7, CEN TC, ISO, and other achieve the connectivity between local systems.

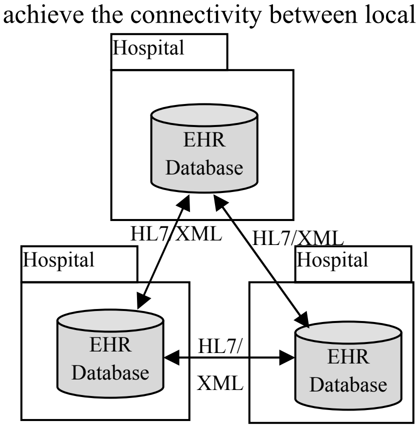

Fig. 3: Distributed universal EHR architecture

This architecture is the most flexible one, and it is used in our model.

2.3 EHR structures

In HER systems, there are four record structures [13]: *(1) Integrated record:* Data are presented in a strictly chronological way, identifying each episode of care by time and date.

(2) Source-Oriented Medical Record (SOMR): It is organized according to the source that generated the data, typically different hospital departments.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

(3) Protocol-driven patient record: In this model, a template is captured in a pre-structured form that dictates what specific data are to be obtained and recorded by the clinician.
(4) Problem-Oriented Medical Record (POMR): It organizes data according to the list of patient problems. The problem list acts as an index to the whole record. All progress notes, tests, treatment notes and medications are numbered according to the problem they relate to. Progress notes are often written according to the Subjective Objective Assessment Plan SOAP template.
No single patient record structure will suit every purpose. In our model will support POMR because physician concentrates on patient problems when searching for and storing data.

2.4 EHR standards

We concentrate on three EHR design problems where are (1) Healthcare messages exchange, (2) EHR object model (i.e. content and structure), and (3) Healthcare terminology/ vocabulary. In addition, the EHR system should incorporate a proper security mechanism.
Table 1 [12] summarizes how the three standards organizations support the three parameters of EHR.

Table 1: EHR Standards [12]

	HL7	CEN TC 215	ASTM E31
Messaging	HL7 V3	ENV 13606 - 4	ASTM E1238, ASTM1394, ASTM E1467
EHR Object Model	no standard	partially: ENV 13606 - 1, ENV 13606 - 2	partially: ASTM 1384
Terminology	LOINC, SNOMED, UMLS, etc.	no standard	ICD9, SNOMED, etc.

The table shows that while standards for healthcare message exchange and terminology exists, there is no standard for EHR Object Model. HL7 [14] is in the initial phases of defining EHR content. CEN TC 215 suggests an EHR structure but does not define its content. ASTM E31 is the closest to an EHR object model [3], but it still lacks a lot of essential information and it's not clear how to extend the given model. As a result our proposed model tries to handle this shortage and develop a generic EHR data model.

3. Problem Statement

EHR integrates heterogeneous data from many sources which is important for knowledge discovery. The complexity, sparseness and fast evolution of clinical information makes development and maintenance of clinical databases challenging [18]. Conventional (relational) databases have static design. On the other hand, in healthcare environment, entities and attributes (physical design) are changed continuously. This can be time consuming for the maintenance and upkeep of the database and all applications depending on it and user interface must be changed as well. Also, in relational database there are a maximum number of attributes for each relation and a set of predefined data types which cannot be extended. EHR databases should ideally be flexible enough to handle new data types and attributes without needing to change the physical database schema.
Also, Time is important in EHR data model. Representing, maintaining, querying, and reasoning about time-oriented clinical data are critical success factor. Although EHR contains temporal data as patient history, the structure of conventional database does not make it easy to store time-varying data.
Healthcare data are sparse. Each entity will have thousands of attributes, but each row in the table will use only small set of these attributes as patient tests. This situation waste storage space as shown in table 2.

Table 2: EHR sparseness of data

Id	name	X	Y	Z	A	B	C	..	D
1	N1	x1	null	null	a1	b1	null	..	null
2	N2	null	null	z2	null	null	c2	..	d2
3	N3	null	y3	null	null	b3	null	..	d3

We require an open schema data model to support dynamic schema change, sparse data, temporal and high dimensional data. In open schema data models, logical model of data is stored as data rather than as schema, so changes to the logical model can be made without changing the schema. In this paper, we will use relational design to build EHR data model after solving its problems and handling its shortages. The problems as sparseness, continuously evolving problems [20, 22], adding new data types, and temporal data modeling [4, 19] are solved in our model. In this model, we use EAV/CR modeling when we need to model sparse and dynamic data. Non-sparse data as patient demographic data stored in conventional relational tables. This is a mixed-schema design. Also, EAV permits the extraction of archetypical data from database to facilitate data interoperability [21]. For interoperability and knowledge discovery purposes, there are also vocabulary tables as shown in figure 4 [22].

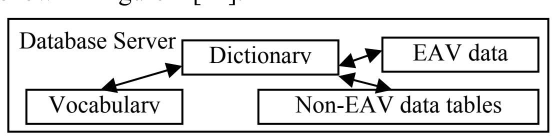

Fig. 4: EHR database server.

4. Proposed Data Model

In this section, we will propose our data model for EHR database that solves all of its design problems.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

4.1 Evolution of EAV and EAV/CR

The model allows the storage of conventional relational data for not sparse data. Also, if new data type is needed, class diagram can be defined and map any class to relational table or to a type of column. Spare data will be stored in EAV sub-schema. This behavior is used to minimize the negative effects of EAV to the whole EHR system as shown next. At the same time, it meets the challenges in section 3.
In relational databases, the logical and physical schemas have the same layout. The complexity and the rapid evolution and expansion of the domain of clinical information require a large maintenance overhead if data are laid out using a relational design. For this reason, it is desirable that EHR database be flexible and allow for modifications and for addition of new types of data without having to change the physical database schema. The ideal design would implement a highly-detailed logical database schema in a completely-generic physical schema that stores the wide variety of clinical data in a small (and constant) number of tables.

- *Basic EAV Schema*
In EAV design all data can be stored in a single generic table with conceptually 3 columns: 1 for entity (e.g., patient identification), 1 for attribute (e.g., name), and 1 for value (e.g., "Jens Hansen") [28]. To add more descriptive fields to the entity class, you add attribute values in the attribute field as in table 3, 4. The main advantages of this design are flexibility and effective entity-centered queries and storage saving by preventing fields with NULL values.

Table 3: Conventional relational database

Patient_id	Date	Name	Test1	Test2
1	1/1/2012	Hans	T1	null
2	2/1/2012	Gems	T11	T22
3	3/1/2012	Tom	null	T222

Table 4: EAV database design

Patient_id	Date	Attribute	Value
1	1/1/2012	Name	Hans
1	1/1/2012	Test1	T1
2	2/1/2012	Name	Gems
2	2/1/2012	Test1	T11
2	2/1/2012	Test2	T22
3	3/1/2012	Name	Tom
3	3/1/2012	Test2	T222

In EHR environment, the meaning of null (inapplicable, unavailable or unknown) is sometimes required. To handle this situation, a "missing value code" column should be added to an EAV table, which is non-null only when the value column is null. The main disadvantages are data display, attribute-centered query complexity and inefficient constraint checking. The logical schema is described in EAV metadata relational tables, so any change to the logical structure will be added, deleted, or updated without affecting the physical structure. Metadata plays a critical role in virtually converting EAV physical schema to the relational-like, user friendly logical schema. This allows end users to query EAV as relational data.
The non-sparse data as patient demographics can be modeled in conventional tables and connect this table to EAV table that model sparse and evolving data, figure 5.
In figure 5 the EAV data table contains:

- The entity. For clinical findings, the entity is the patient event: a foreign key into a table that contains at a minimum a patient ID and one or more time-stamps (e.g., the start and end of the examination date/time) that record when the event being described happened.
- The attribute or parameter: a foreign key into a table of attribute definitions (in this example, definitions of clinical findings). At the very least, the attribute definitions table would contain the following columns: an attribute ID, attribute name, description, data type, units of measurement, and columns assisting input validation, e.g., maximum string length and regular expression, maximum and minimum permissible values, set of permissible values, etc.
- The value of the attribute. This would depend on the data type.

Querying EAV data is more difficult than conventional data as follows: querying tables 4, 5 for patient 'Hans' using SQL:

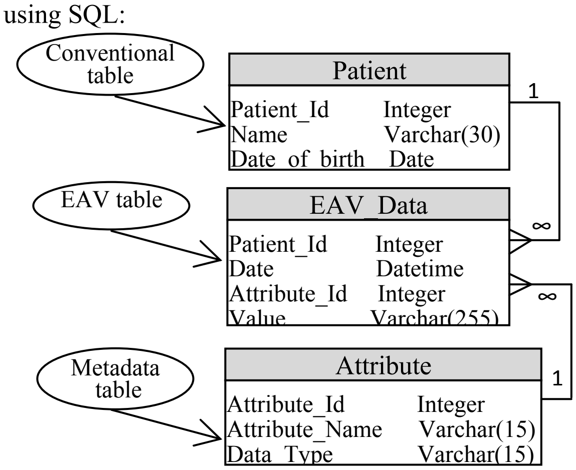

Fig. 5: Simple EAV schema

Table 4:

```
SELECT *
FROM table4
WHERE name LIKE 'Hans%'
  AND Date < '2/1/2012';
```

Table 5:

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

```
SELECT d1.patientID AS patientID,
       d1.value AS name,
       d2.value AS Date
FROM table5 AS d1 INNER JOIN table5 AS d2
  USING (patientID)
WHERE d1.attribute='name'
  AND d1.value LIKE 'Hans%'
  AND d2.attribute = 'Date'
  AND d2.value < '2/1/2012';
```

Querying EAV data includes a self-join for each attribute. This query complexity is solved by:

(1) There may not be a problem. Attribute-centered queries are important for research questions only.

(2) Any need for regular cross-patient data access could be met by making backups of the production database and restoring them onto separate hardware. Resource-intensive queries run on the backup data will not affect the production server. Additionally, the EAV data schema could be transformed into numerous conventional tables after backup thus easing query design by end users with modest SQL skills.

(3) Optimization of queries may increase the efficiency considerably.

(4) EAV data may be pivoted [27] and converted to logical relational model using data warehousing, materialized views or in-memory data structures as hash tables and two-dimensional arrays.

(5) Extension of the SQL language to facilitate "pivoting" of attribute-centered data into a conventional layout—the Extended Multi-Feature (EMF) SQL—can solve the problem.

- *Multi data types EAV Schema*

Values may of course be of any type, for example, text, number, or Boolean (true/false). In the example in Table 5, the value field of the data table is text type. Such a design achieves simplicity by storing all simple types as text values. This approach has, however, some drawbacks. First, not all data types will fit into a text field as Binary objects. Second, queries based on values will be less efficient for non-textual values. The text "12" is less than the text "2" even though it is numerically greater, because text is sorted character by character, from left to right.

One enhancement to EAV design is done by storing data with different data types in separate tables [26]. This multi-data type EAV design allows the use of any data type including text, integer, floating point, time stamp, object and graphical data (Binary Large Object (BLOB)) as in figure 6. Indexing is possible in this design.

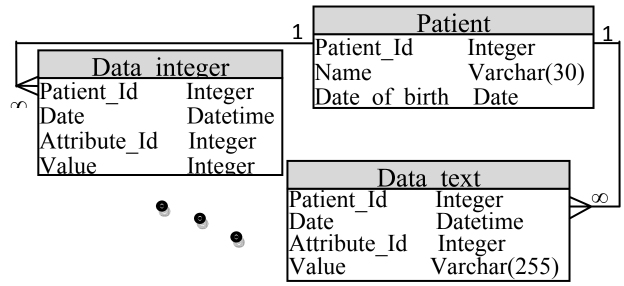

Fig. 6: Multi-data types EAV schema

- *Hybrid EAV Schema*

The best enhancement is hybrid EAV format that allows the use of multiple data types in a single table to provide simpler and faster queries [29], figure 7. It combines the best features of the simple and multi-data type EAV designs without increasing the size of the data storage. *This approach will be used in our model.*

- *Enhancing EAV to EAV/CR*

EAV is convenient when attributes are of simple or primitive data types. However, it is possible that attribute is of object (class instance) type that has substructure. That is, some of its attributes may represent other kinds of objects, which in turn may have substructure, to an arbitrary level of complexity. EAV/CR schema supports complex substructure [30].

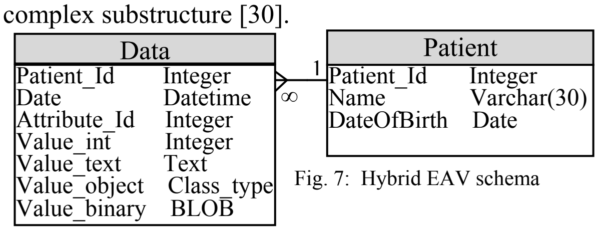

Fig. 7: Hybrid EAV schema

Furthermore, EAV makes it easy to model the one-to-many relation between *patient* and *clinical events*. In EHR, it should be possible to record relationships between *clinical events* (e.g., infection leads to a course of penicillin or myocardial infarction leads to death). EAV/CR schema handles the complex relationships between classes. This model adds an object-oriented framework to the EAV model by definition of classes and relationships. EAV/CR data and metadata are stored in Object-Relational data model. Figure 8 shows the EAV/CR metadata relational tables.

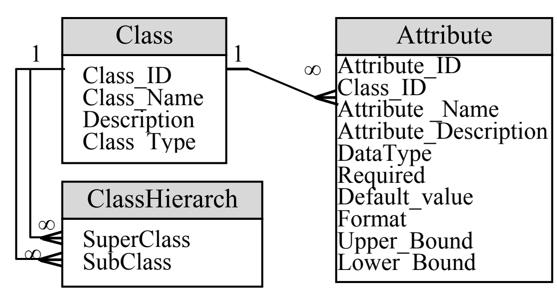

Fig. 8: EAV/CR Metadata relational model

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

In Figure 8, inheritance is designed in the ClassHierarch table. Figure 9 shows how to model objects using EAV schema. Composition relationship is modeled in EAV_Object table where one attribute of the main object can be object of a different class. Also, objects as clinical events can be related in EAV_Object by adding attributes as O1 *in_response_to* O2, or O1 *result_from* O2. Keyword table is used to enhance the search process (Google-style search), by linking some descriptive words to specific objects.
Ad hoc relationship between objects (e.g., penicillin leads to rash) may be recorded as objects themselves. For this purpose, a class ObjectRelation could be defined with two attributes, objectID and relatedObjectID. More descriptive attributes may be added to this class if required—e.g., causality.
If you have hundreds or thousands of non-sparse objects for a given class, you should create a distinct table for this class, preferably with the same name as the Class itself. (Naturally, such tables are not illustrated in the schema above, since they are specific to your application.) The summary information for each object, however, still lies in the Objects table. This way, the Keyword table lets users locate Objects of interest irrespective of what Class they belong to. When you use a Mixed Design approach, where some data is stored in regular tables and other data in EAV form, the metadata in the Class table above should have a flag that states whether Object data for a given class is recorded in conventional columnar form or in EAV form. This way, your application code knows what to do when the user wishes to inspect the details of a particular object.
Obviously, considerable up-front programming is required to drive an ergonomic user interface for the EAV/CR model in a real-life production environment. On the other hand, this is a one-time-only job.

4.2 Data transformation using domain dictionary and rule base

For interoperability, the EHR database must contain standard medical terminologies as UMLS or SNOMED. For knowledge discovery, the medical data have to be transformed into a suitable transaction format. For categorical data, each value is connected to one code. For continuous attributes, we use a rule base such as if age>20 and age < 30 then age=2 and so on [32].

4.3 EHR database is bitemporal

Time is one of the most important variables in healthcare, as medical temporal reasoning is directly relevant to almost every aspect of medical practice. Longitudinal data are essential for outcome analysis, CDSS and temporal data mining. History-taking, causal diagnostic explanation, diagnostic decisions, and patient monitoring all rely heavily on the patient's past clinical course, current medical status and future course (patient treatment).
Modeling time is varying and based on the choice of ontological primitives, whether time is viewed as discrete or continuous, modeling of uncertainty, incompleteness and impreciseness and granularity of the time stamp chosen. The temporal database may store transaction time (*rollback database* and DBMS support that), valid time which is the time when the event is valid (*historical database*) and both (*bitemporal database*).
The EHR bitemporal database must allow:
(1) Relating situations to a calendar.
(2) Relating situations to “reference” situations.
(3) Relating events together in “before- and after-” chains.

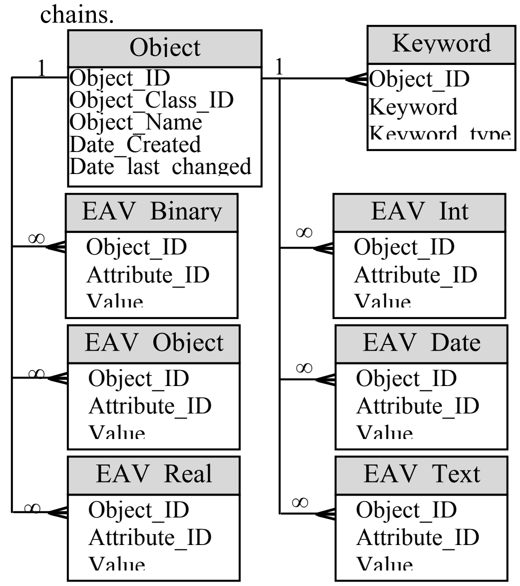

Fig. 9: EAV/CR data model

Moreover, the time of an event may be known with respect to a specific calendar date, time or timestamp (time point or instant) [34], with exact interval (start time and end time, or occurrence time and duration), with respect to the occurrence of some other event, with uncertain time, and/or uncertainly with other event. There are 49 relationships between time intervals and time points [33].
Temporal data model need to manage uncertainty of events, both to actual date/time of occurrence of an event time (e.g., Event A occurred about 5 years ago, and it was winter, around 02:00 AM.), and to the occurrence relative to some other event, both qualitatively (e.g., Event A occurred before Event B) and quantitatively (e.g., Event A occurred approximately 1 week before Event B). Our model will not handle the uncertainty.
The precise time of an event, either time point or time interval, will be modeled by two attributes Event_Start_Time and Event_End_Time. This time

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

interval will have the same value in its both ends in case the event in time point event.

In addition to defining an event with respect to an absolute date/time, the model enables defining the time of an event with respect to the time of some other event. We have 49 potential temporal relationships between two events. This requirement will be achieved by defining another relation named Temporal_Relationships which normalize the many-to-many temporal relationship between two events as shown in Figure 10.

The temporal relationship between any two events can be described by the attribute Temporal_Relationship. The domain for this attribute is the 49 temporal relationships. In addition to these 49 qualitative relationships (e.g., Event A occurred before Event B), the model represents quantitative temporal relationships (e.g., Event A occurred approximately 2 weeks before Event B) by using the attribute Relative-Interval, which represents the amount of time between two related events.

There are many query languages which extends SQL as Time Line SQL (TLSQL), TSQL2, TQuel and other which facilitate the querying of the temporal database.

Domain expert must choose the appropriate level of granularity for each table which may be one of:

Years: <year>
Months: <year, month>
Days: <year, month, day>
Hours: <year, month, day, hour>
Minutes: <year, month, day, hour, minute>
Seconds: <year, month, day, hour, minute, second>

The interval will contain a finite sequence of discrete, indivisible *time quanta*, according to granularity. If the Days level is chosen, duration of five means five days.

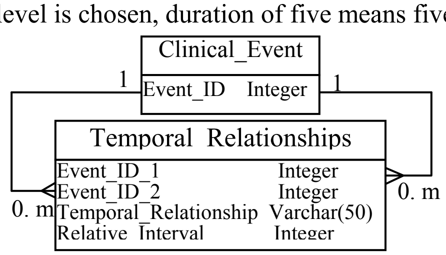

Fig. 10: Management of temporal relationships

4.4 Details of proposed model

The proposed framework tries to model all patient clinical events. To keep the simplicity of the model, standardization tables such as medicine type, controlled vocabulary for problem list (SNOMED CT) and terminology tables are removed, sparseness (EAV tables) will be designed aside, and only attributes required for relationships between entities (primary key) are modeled (entities contains other attributes). The conceptual schema should correspond to expert understanding of the medical domain. It should be concerned with data representation and not with issues of data storage and access efficiency. These sub-schemas can be appended easily to our schema. Our model is problem oriented temporal model where patient problems are the core component. This structure provides a means of depicting a patient's problems and all clinical entries related to each problem (e.g., symptoms, physical findings, laboratory tests, etc.). This grouping of data provides a context within which one may interpret clinical information. EHR also contains summary data from other systems. When detailed data are needed, a link to these systems (RIS, PACS, LIS and PIS) can be added.

The clinical heart of the EHR is the core of the entities: patient, provider, problem, encounters, orders, services and observations. Representing the overall structure of the record is difficult since it is complex and has a number of dimensions. It also can be viewed from many perspectives. Four of these are: chronological, by encounter/episode, by problem, and by topic. Each of these views looks at the same stored data in a different way. Our model concentrates on patient problem list as in Figure 12. The model starts with the concept of a person's clinical file which is patient identification, social and demographics data as in Patient table (non-sparse table). The model allows an unlimited number of identifiers for patient—a primary key for internal consistency plus any number of external identifiers [23]. It allows an extensible set of identification values to use for both ID lookups and de-duplication requirements that crop up when integrating multiple systems. Tables Person_Identifier and Identity_Type are used to achieve this purpose. The SSN and Health Record Number are rows in the Identity_Type table and the Patient_Identifier table references the appropriate identity_type_id (for example, SSN) and store the actual type value in the Identifier column. This model allows having as many different identities as necessary into the future. Patient table stores demographic data (e.g., names, sex, birth date, death date, etc), social data (Father, mother, and family IDs) to build social network and allow Social Network Analysis and Mining (SNAM), and linked to Address table which store patient detailed addresses over time. The addresses data can be used in spatial data mining.

Table Vital_Sign stores complete vitals over time (time is part of primary key), such as height, weight and blood pressure for each patient. This table stores temporal data so attributes as Start_date_Time and End_date_Time are required. For the most recent vitals, the End_Date_Time is NULL. Each person has a set of problems, the Problem table. Problem table is sparse, so we have to extend it by EAV_Problem table as shown next. The problems are identified by the physician, Physician table. Problems may also have temporal relationships with each other and this is handled by Temporal_Relationships table. Each problem has Status (Active/Inactive /Resolved/Erased), Link to the

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

previous problem, Start_Date_Time and End_date_time. For the current patient problems, the End_date_time is NULL. In table 5, problems P2, P3 are current problems, and P2 reference P1 (P2 occurs after P1) and P3 reference P2. The model allows one problem to be linked to many problems, and many problems to link one problem simply be conventional table Problem or by EAV_Problem table. This feature allows us to record the evolution of each problem until it is resolved.

Each problem has one initial plan (Problem_Initial_Plan) which is a set of structured data that must be recorded whenever such problem is identified in a patient.

The evolution of a patient's problem is recorded through its EPISODES, the set of which is called the PERSON PROBLEM'S DIARY. In an outpatient clinic environment, each episode of a problem corresponds to the occurrence of an encounter; in each ENCOUNTER several problems can be addressed, hence several EPISODES can be created. For each EPISODE several pieces of information that correspond to the proposed Subjective, Objective, Assessment and Plan (SOAP) structure for progress notes can, optionally, be recorded or created. Each episode has Prescription which has a set of Medications which describes the types of medicine the patient will take for the problem. Observation table stores the patient's assessments and diagnoses. It stores the observations of the physician during structured and systematic examinations of the patient's body during encounters/episodes. It allows characterization of expressed problems with observational evidence in explicit common terms and measures that, over time, will allow practitioners to follow the course of illness and recovery. Laboratory tests are stored in Lab_Test table, and radiological tests in Radiology_Procedure.

Each problem has many lab and radiology tests and the tables contain summary information about it. If clinician needs to see test details, the model contains a link to the RIS or LIS. Some problems may need non medical therapies which may be advices or practices. These data are stored in Non_Medical_Therapies table. In an encounter the physician may need to schedule a set of events for the patient. Each problem may have a treatment plans and orders (Order/Plan table). A care treatment plan may be a broad perspective program that identifies planned clinical encounters, education and scheduled events (Scheduled_Events table) related to specific diagnosis or problem. Each order may require a set of clinical services. Electronic ordering should include medications, oncology management, laboratories and radiology. Orders should automatically be routed to the department and staff responsible for executing them. The class hierarchy in Figure 8 will be added to the model, and any complex type can be defined as a class hierarchy and used as table or column type. Also, the EAV/CR can extend any sparse table as Problem table. Any table for EAV modeling will begin by "EAV_". Figure 11 shows how to extend conventional relation as Problem with the dynamic attributes in EAV_Problem without affecting the physical design of database. This process requires domain expert to define each table's conventional part. EAV_Problem table contains Value_Missing attribute to record the absence of value for attribute if it makes sense.

Table 5: A Problem table snapshot

ProblemID	Name	Status	Link	Start	End
1	P1	A		1/1/2012	1/3/2012
2	P2	A	1	2/3/2012	null
3	P3	A	2	2/3/2012	null

The same extension is done for Episodes table. Moreover, the referential integrity constraint is preserved in this extended model because the relationships are between the conventional tables and the integrity which is preserved by the DBMS.

Many types of relationships exist in our model:

1- Inheritance (between classes).
2- Composition (if class define attribute as object or if EAV table has attribute of type object).
3- The conventional design relationships.
4- The relationships between EAV tables and conventional tables.
5- The temporal relationships.

If any entity has temporal relationships with other entity including itself, it can use the ideal in Figure 10. We will not represent all these relationships in the model to preserve simplicity.

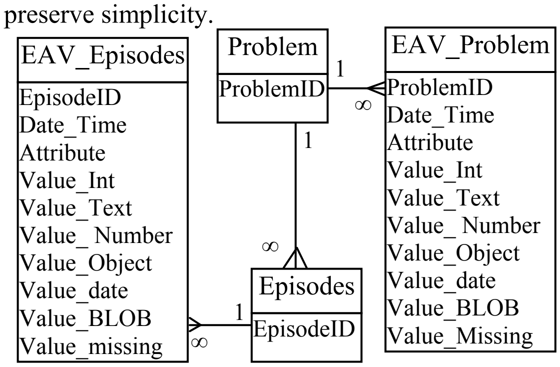

Fig. 11: EAV/CR Extension with referential integrity preserved.

5. SCOPE AND LIMITATIONS

This EHR data model can be applied in any healthcare organization which needs to enhance its patient care capability. The model can preserve a complete picture about patient's current and past medical and clinical conditions. It is an open model and allow designer to select which attributes to add. At the same time the model optimize memory usage. Because of space

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.
IJCSI
www.IJCSI.org

restrictions, the model needs to handle the interoperability issues. If the hospitals need to build a national EHR, the communication between local EHR must be handled may be by using a standardized terminology server as SNOMED CT.

6. Conclusion

In this paper, we proposed a generic, clinical and temporal data model for the EHR database using object relational data model. This is achieved by using a mixed design consisting of standard and generic tables so that varied data types can be represented in the database. Because patient problems are the most important driver for physicians' queries in healthcare, this model models a problem oriented record. This logical design did not present data attributes for entities to preserve the simplicity and to make the design as generic as possible.
The most important entities are modeled and any temporal or generic entity can be modified with minimum effect on the physical design of the database. This solution allows continuous modifications of EHR database requirements to be applied smoothly and with minimum effect on EHR system applications. The temporal, social, and clinical aspects of the model enhance the data mining and knowledge discovery processes on the EHR database which enhances the operations of CDSS. The next step will be to implement this logical model and apply data mining techniques on its data. We will solve the interoperability problem in the next version of the model. Moreover, we will build a clinical decision support system. This system collects patient data from EHR and takes clinical decisions to aid healthcare personnel to make clinical decisions.

References

[1] A. Begoyan, "An overview of interoperability standards for electronic health record", Integrated Design and Process Technology, Society for Design and Process Science (IDPT), 2007.

[2] International Organization for standardization (ISO) "Requirements for electronic health record reference architecture", ISO TC 215/SC 2002.

[3] American Society for Testing and Materials (ASTM), "ASTM E1384 - 07 Standard Practice for Content and Structure of the Electronic Health Record (EHR)", Available from: URL: http://www.astm.org/Standards/ E1384.htm, last seen: January 2012.

[4] European Committee for Standardization CEN/ TC251 preENV 12381, "TSMI: a standard for time specific problems in healthcare informatics and telematics 1997," www.cen.eu, last seen: Jan. 2012.

[5] W. MacKinnon, M. Wasserman "Integrated Electronic Medical Record Systems: Critical Success Factors for Implementation," IEEE Proceedings of the 42nd Hawaii International Conference on System Sciences, On Page(s): 1-10, issue date Jan. 2009.

[6] W. Xu, Z. Guan, H. Cao, H. Zhang, M. Lu, T. Li; "Analysis and evaluation of the Electronic Health Record standard in China: A comparison with the American national standard ASTM E 1384," international journal of medical informatics, Vol. 80, Issue 8, Pages e39-e188, 541-606, 2011.

[7] R. Elmasri, "Fundamentals of Database Systems (6th Edition)," Publisher: Addison Wesley, ISBN-13: 978-0136086208, 2010.

[8] P. Nadkarni, L. Marenco, R. Chen, E. Skoufos, G. Shepherd, DPhil, P. Miller; "Organization of Heterogeneous Scientific Data Using the EAV/CR Representation," J. of the American Medical Info-rmatics Association JAMIA / 6(6):478-93, 1999.

[9] M. Luis, T. Nicholas, C. Chiquito, S. Gordon, M. Perry, N. Prakash; "Achieving Evolvable Web-Database Bioscience Applications Using the EAV/CR Framework: Recent Advances," Journal of the American Medical Informatics Association 10 (5): 444–53, 2003.

[10] D. Valentin, N. Prakash; "Guidelines for the effective use of entity-attribute-value modeling for biomedical databases", International journal of medical informatics 76 (11–12): 769–79, 2007.

[11] R. Sriram, "The Role of Standards in Healthcare Automation," 5th Annual IEEE Conference on Automation Science and Engineering Bangalore, India, August 22-25, On page(s): 79 – 82, 2009.

[12] S. Cohen, A. Shabo; "Electronic Health Record (EHR) Standards Survey," IBM research, 2001, Available at: http://www. research.ibm.com /haifa/projects/software/imr/papers/EHRSurvey.pdf , last seen: January 2012.

[13] E. Coiera, "Guide to Health Informatics," Publi-sher: Hodder Arnold Publishers; 2nd edition, 2003.

[14] Health Level Seven International [Online]. Available from: URL: http://www.hl7.org/ index.cfm/, last view: January 2012.

[15] Z. Yaowei, X. Yabin; "Design of Electronic Medical Record Data Integration Model Based on OGSA-DAI," IEEE 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), on page(s): V3-278 - V3-281, 2010.

[16] C. Kumar, C. Rao, A. Govardhan; "A framework for interoperable healthcare information systems," IEEE International Conf. on Computer Information Systems and Industrial Management Applications (CISIM)," pages: 604 – 608, 2010.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

[17] Frank L., Andersen S.K.; "Evaluation of Different Database Designs for Integration of Heterogeneous Distributed Electronic Health Records," The IEEE/ICME Inter. Conf. on Complex Medical Engineering (CME), page(s): 204 – 209, 2010.

[18] J. Anhoj, "Generic design of web-based clinical databases," Journal of Medical Internet Research, vol. 5(e27), November 2003.

[19] T. Yong, T. Na, Y. XiaoPing, F. ZhiSheng, X. Wei; "A Unified Model of Temporal Knowledge and Temporal Data," The IEEE 8th International Conference on Computer Supported Cooperative Work in Design. Pages: 711 - 713 Vol.1, 2004.

[20] Paul R., A. Latiful Hoque; "Optimized Entity Attribute Value Model: A Search Efficient Representation of High Dimensional and Sparse Data," the 9th Asia Pacific Bioinformatics Conference (APBC2011), IBC, vol. 3, article no. 9, pp. 1-6, 2011.

[21] G. Duftschmid, T. Wrba, C. Rinner; "Extraction of standardized archetyped data from Electronic Health Record systems based on the Entity-Attribute-Value Model," International J. of Medical Informatics; Vol. 79(8); pp. 585 - 597; 2010.

[22] P. Nadkarni, C. Brandt, S. Frawley, F. Sayward, R. Einbinder, D. Zelterman; "Managing attribute-value clinical trials data using the ACT/DB client-server database system," J. American Medical Informatics Association vol. 5, no: 2, 139–151. 1998.

[23] T. Beale, "GEHR System Architectures," www.gehr.org, 2003, last seen: Jan. 2012.

[24] I. Roma´n, L. Roa, J. Reina-Tosina, G. Madinabeitia; "Demographic management in a federated healthcare environment," International Journal of Medical Informatics, Volume 75, Issue 9 , Pages 671-682, 2006.

[25] X. Wang, J. Ling; "Multiple valued logic approach for matching patient records in multiple databases," J. of Biomedical Informatics, pp. 224-230, Vol. 45, 2012.

[26] Gilchrist, J., Frize, M., Ennett, C.M., Bariciak, E.; "Performance Evaluation of Various Storage Formats for Clinical Data Repositories," IEEE Transactions on Instrumentation and Measurement, Vol. 60, Issue: 10, Page(s): 3244 – 3252, 2011.

[27] V. Dinu, P. Nadkarni, C. Brandt; "Pivoting approaches for bulk extraction of Entity–Attribute–Value data," ACM Journal of Computer Methods and Programs in Biomedicine archive, Vol. 82, Issue 1, 2006.

[28] S. B. Johnson, "Generic data modeling for clinical repositories," Journal of the American Medical Informatics Association (JAMIA), vol. 3, pp. 328-339, 1996.

[29] PRAKASH M. NADKARNI, C. BRANDT; "Data Extraction and Ad Hoc Query of an Entity–Attribute–Value Database," Journal of the American Medical Informatics Association, Vol. 5, Num. 6, 1998.

[30] R. S. Chen, P. Nadkarni, L. Marenco, F. Levin, J. Erdos, P. L. Miller; "Exploring Performance Issues for a Clinical Database Organized Using an Entity-Attribute-Value Representation," Journal of the American Medical Informatics Association, Vol. 7 Number 5, pages 475–487, 2000.

[31] R.H. Dolin, "Modeling the temporal complexities of symptoms," Journal of the American Medical Informatics Association, Vol. 2, Num. 5, 1995.

[32] Paul R.; Hoque A.S.M.L.; "A storage & search efficient representation of medical data," IEEE International Conference on Bioinformatics and Biomedical Technology (ICBBT), Page(s): 418 – 422, 2010.

[33] CEN (European Committee for Standardization)/Technical Committee 251-Medical Informatics. Project Team 017. Time Standards for Healthcare Specific Problems. Draft. CEN/TC251/ PTO17. September 30, 1994.

[34] Nadeem M., Aqil B., Kamran A.; "A Logical Temporal Relational Data Model," IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 1, No. 1, 2010.

[35] International Organization for Standardi-zation (ISO), Available: http://www.iso.org/iso/iso _technical_committee?commid =54960. Last seen: January 2012.

[36] International Organization for Standardi-zation (ISO), Available: http://www.iso.org/iso/ iso_catalogue/catalogue_tc/catalogue_detail.htm?cs number=39525. Last seen: January 2012.

[37] International Organization for Standardization, [Online]. Available: http://www.iso.org.

[38] Digital Imaging and Communications in Medicine, [Online]. Available: http://medical.nema.org.

[39] The European Committee for Standardization, [Online]. Available: http://www.cen.eu.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.